

# Efficiently storing and reshaping large data

Hadley Wickham  
Rice University

# Motivation

	Identifier	Measured
aka	dimension, key, index of rv	measure, random variable
values	fixed by design	measured in expt
type	consecutive integers (wlog)	anything

# Storage

x	y	v
0	0	a
0	1	b
0	2	c
1	1	e

data frame

x	y	v
0	0	a
0	1	b
0	2	c
1	1	e

data frame

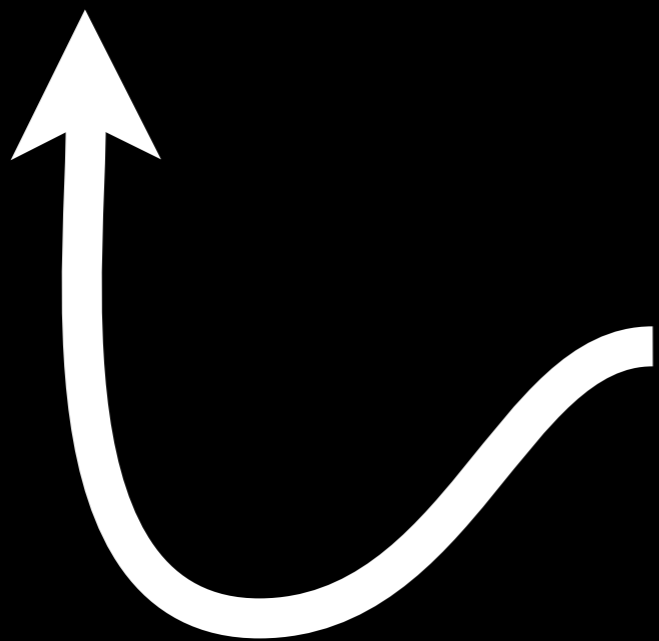
a	.
b	e
c	.

2d array

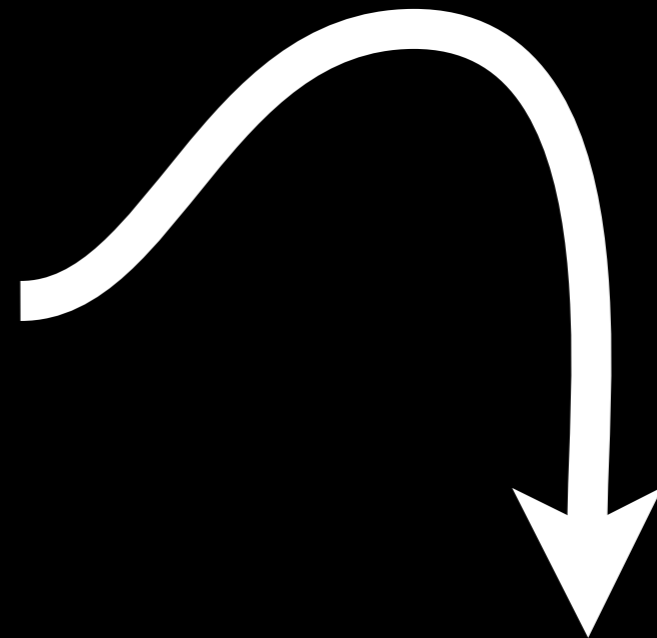
a	.	b	e	c	.
---	---	---	---	---	---

row major

C



a	.
b	e
c	.

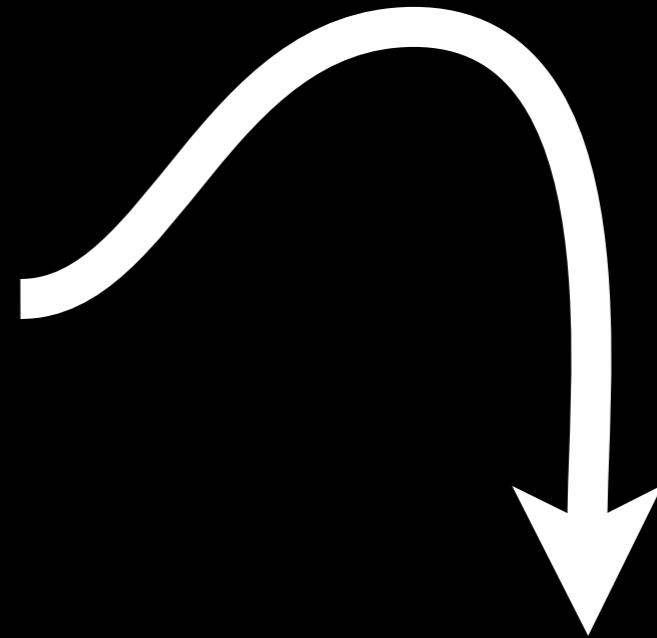


fortran, R, matlab

column major

a	b	c	.	e	.
---	---	---	---	---	---

a	.
b	e
c	.



**column major**

a	b	c	.	e	.
---	---	---	---	---	---



i	x	y	v
0	0	0	a
1	0	1	b
2	0	2	c
3	1	0	.
4	1	1	e
5	1	2	.

$$x = i // 3$$
$$y = i - 3 \cdot x$$

[en.wikipedia.org/wiki/Row-major\\_order](http://en.wikipedia.org/wiki/Row-major_order)

# Storage

- Assuming no missing values
- Data frame =  $(p_i + p_m) \cdot n$
- Matrix =  $p_m \cdot n$
- Data frame excess =  $1 + p_i / p_m$

# Missing values

- Missing values make a **big** difference.
- **Id** variables: Structural missings from experimental design. If no crossing then data frame only requires  $\max(m_1, m_2, m_3)$  but matrix requires  $m_1 \cdot m_2 \cdot m_3$
- **Measured** variables:  $x\%$  MCAR allows data frame to drop  $x^{pi}$  rows on average

	Data frame	Matrix
Missings	implicit	explicit
Id variables	explicit	implicit
Best for	nested data	crossed data
Dimensionality	2	$n (l)$
Memory	scattered	contiguous

var	lat	long	time	value
ozone	-21.2	-113.8	1	260
ozone	-18.7	-113.8	1	258
ozone	-16.2	-113.8	1	258

7	24	24	72	232-
---	----	----	----	------

209,304 values: data frame: 11.3 meg, matrices: 2.2 meg

# Reshaping

# Reshaping

- Many possible forms of data useful for different types of data analysis
- Most statistical algorithms compare columns of data frame or 2d matrix
- To reshape: construct new ordering of values, then set dimensions

# Output

- Two extremes:
  - $p_i$ -d matrix =  $m_1 \sim m_2 \sim m_3 \sim m_4$
  - $n \times p_i$  data frame =  $m_1 + m_2 + m_3 + m_4$
- Many possible (useful) intermediate forms in the middle
  - $m_3 + m_2 \sim m_1 + m_4$  etc
- How do we go from one form to another?



# Reshaping

- Turns out to be fairly simple - we just need to work out the right linear ordering
- For each output dimension, create a single new variable
- Create overall order from individual dimensions, filling in with missings as needed

a	b	c	val
0	0	0	5
1	0	0	10
2	0	1	15
0	1	1	20
1	1	2	25
2	1	2	30

$$c + a \sim b$$

$$c + 3 \cdot a$$

c	a	d1
0	0	0
0	1	3
1	2	7
1	0	1
2	1	5
2	2	8

b	d2
0	0
0	0
0	0
1	1
1	1
1	1

$$d_1 + 8 \cdot d_2$$

$d_1$	$d_2$	ov	val
0	0	0	5
3	0	3	10
7	0	7	15
1	1	9	20
5	1	12	25
8	1	16	30



	0	1
0	5	.
1	.	20
2	.	.
3	10	.
4	.	.
5	.	25
6	.	.
7	15	.
8	.	30

# Aggregation

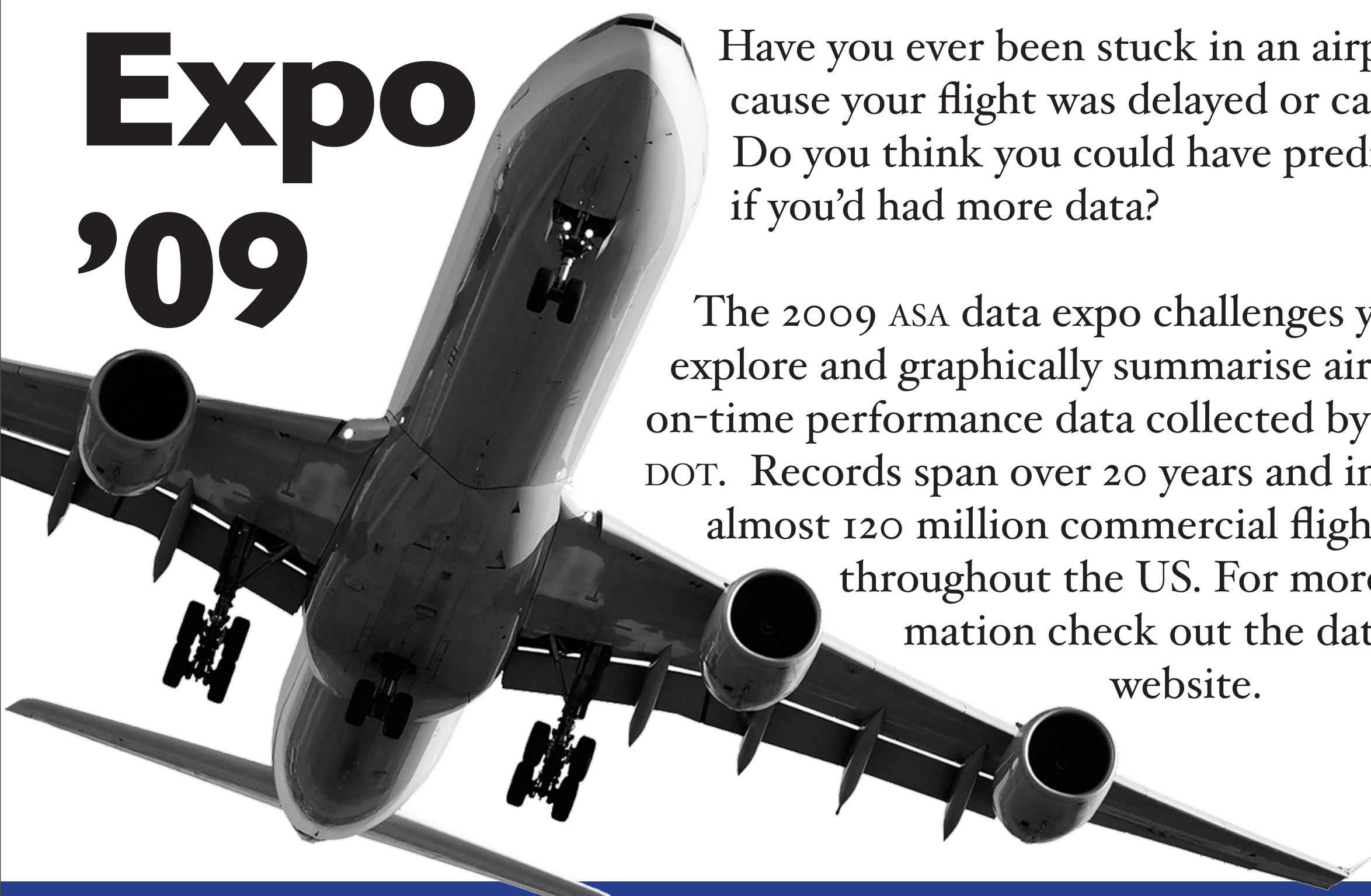
- The process for aggregation is much the same, but the overall value will no longer be unique
- Use aggregation function to collapse to a single number

# Performance

- For very large data, often just want to work on a subset
- Need on-disk storage and efficient access
  - Sorting
  - RDMS (but usually don't update)
  - Compressed bitmap indices:  
<http://sdm.lbl.gov/fastbit/>

# ASA Data

# Expo '09



Have you ever been stuck in an airport because your flight was delayed or cancelled? Do you think you could have predicted it if you'd had more data?

The 2009 ASA data expo challenges you to explore and graphically summarise airline on-time performance data collected by the DOT. Records span over 20 years and include almost 120 million commercial flights flown throughout the US. For more information check out the data expo website.

<http://stat-computing.org/dataexpo/2009>